ERDC/CERL SR-05-59

Construction Engineering Research Laboratory

US Army Corps of Engineers®
Engineer Research and Development Center

Procedure for Estimating Weights of LEAM

User's Guide

Shoufan Fang, James Westervelt, George Z. Gertner, and Alan B. Anderson

December 2005

ERDC/CERL SR-05-59
December 2005

Procedure for Estimating Weights of LEAM: User's Guide

James Westervelt and Alan B. Anderson

Construction Engineering Research Laboratory
PO Box 9005
Champaign, IL 61826-9005

Shoufan Fang and George Z. Gertner

University of Illinois
Department of Natural Resources and Environmental Sciences
1102 South Goodwin Avenue
Urbana, IL 61801

Final Report

Approved for public release; distribution is unlimited.

Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Work Unit #LHB740 (CNN-T612FF)

ABSTRACT: This document describes spatially explicit methodology to calibrate the Land use Evolution and impact Assessment Model (LEAM) using historic land use data. The LEAM model is used to predict urban growth and environmental impacts associated with urban growth. The model uses physical and social factors to simulate the dynamics of urban development. The Procedure for Estimating the Weights of LEAM (PEWL) uses logistic regression techniques and historic land use data to statistically calibrate the 13 subfactors used in LEAM. PEWL consists of computer programs to prepare samples, generate programs for regression analysis, and produce data files (estimated weights) for simulation using the Urban Sprawl Submodel of LEAM. The generated analysis programs from PEWL are executed using SAS, a common statistical package, to complete logistic estimation. PEWL was developed in FORTRAN language on Microsoft™ Developer Studio (FORTRAN PowerStation 4.0, 1993-1994) software. PEWL includes both the source and executable files of the FORTRAN programs. For demonstration purposes, the approach is demonstrated for a multi-county area around Fort Benning, GA.

Contents

List of Figures and Tables

Figures

Tables

Equations

Acronyms

CERL	Construction Engineering Research Laboratory
ERDC	Engineer Research and Development Center
GIS	Geographic Information System
LEAM	Land use Evolution and impact Assessment Model
SERDP	Strategic Environmental Research and Development Program

Preface

This study was conducted for the U.S. Army Corp of Engineers under 622720A896 “Error and Uncertainty Analysis in Multi-factor Analysis,” LHB740 (CNN-T612FF). The technical monitor was Dr. William D. Severinghaus, CEERD-CVT.

The work was performed jointly by the Ecological Processes Branch (CN-N) of the Installations Division (CN), Construction Engineering Research Laboratory (CERL) and the University of Illinois at Urbana-Champaign (UIUC). The CERL Principal Investigator was Alan B. Anderson. The UIUC Principal Investigators were Dr. Shoufan Fang and Dr. George Gertner. The technical editor was Linda Goersch, Information Technology Laboratory — Champaign. Alan Anderson is Chief, CN-N, and L. Michael Golish is Acting Chief, CN. The associated Technical Director is Dr. William D. Severinghaus. The Acting Director of CERL is Dr. Ilker R. Adiguzel.

CERL is an element of the U.S. Army Engineer Research and Development Center (ERDC), U.S. Army Corps of Engineers. The Commander and Executive Director of ERDC is COL James R. Rowan, EN, and the Director of ERDC is Dr. James R. Houston.

1 Introduction

Background

The LEAM Model

This document describes and demonstrates a quick and inexpensive approach to calibrate the Land use Evolution and impact Assessment Model (LEAM) using historic land-use data. LEAM is used to predict urban growth and environmental impacts associated with urban growth. It uses physical and social factors to simulate the dynamics of urban development.

The Procedure for Estimating the Weights of LEAM (PEWL) uses logistic regression techniques and historic land-use data to statistically calibrate the 13 subfactors used in LEAM. PEWL consists of computer programs to prepare samples, generate programs for regression analysis, and produce data files (estimated weights) for simulation using the Urban Sprawl Submodel of LEAM. The generated analysis programs from PEWL are executed using SAS, a common statistical package, to complete logistic estimation. PEWL was developed in FORTRAN language on Microsoft® Developer Studio (Fortran PowerStation 4.0, 1993-1994) software. PEWL includes both the source and executable files of the FORTRAN programs. For demonstration purposes, the approach is demonstrated for a multi-county area around Fort Benning, GA.

Land-use Conflicts

Incompatible land-use conflicts that affect military installation training and testing are developing as the population of the United States continues to grow and communication and transportation technologies support more diffused settlement patterns (Goran et al. 2000). Various techniques have been developed and applied to predict the potential for incompatible land uses developing around military installations (Deal 2002; Deal et al. 2002; Orland et al. 2001). Techniques range from simple questionnaires answered by installation personnel to spatially explicit dynamic simulation modeling. Perhaps the most common approach for predicting the potential for future incompatible land use involves applying 1-, 3-, and 5-mile* buffers around installations using current and historic digital land-use maps. By counting the urbanized area in these buffers over a time-series of maps, a simple graph of time vs. total amount of urbanized area can be created (Lozar et al. 2003). These simple trends can be very useful and are often sufficient for many analyses.

The extrapolation of past trends can be misleading, however. Growth might continue to accelerate in situations where land available for development is still plentiful and where the fringe of a major urbanized area is entering the area of interest. Growth might be moderate if an urbanized center is already in the buffer and growing along the edge of an installation. It might decelerate if little or no land is left to develop. To help remove the uncertainty, a more careful analysis of the spatial relationships of growing urban centers, available land, natural (e.g., rivers) or man-made barriers (e.g., limited access highways), and zoning can be used to predict potential incompatible land-use situations. LEAM was developed specifically for this purpose (Deal 2002).

LEAM is a computer-based model that simulates land-use change across space and time. It enables planners, policy-makers, interest groups, and laypersons to visualize and test communal decisions and evaluate potential consequences of those decisions. The LEAM environment is used to enhance our understanding of the connection between urban, environmental, social, and economic systems.

The fundamental LEAM approach to modeling urban land-use transformation dynamics begins with drivers, those forces (typically human) that contribute to land-use change. Each driver is developed as a contextual submodel run simultaneously in each grid cell of raster-based geographic information system (GIS) map(s); linked to form the main framework of the model and produce landscape simulation scenarios. Submodels are completed and run independent of the larger LEAM framework so that variables can be scaled and plotted in formats that help visualize and calibrate submodel behavior before it becomes integrated into the larger model.

LEAM model drivers represent the dynamic interactions between the urban system and the surrounding landscape. These drivers have associated weights that are used to combine the drivers into a probability for urbanization. Scenario maps

* 1 mile = 1.61 kilometers

visually represent the resulting land-use changes. Altering input parameters (different policies, trends, and unexpected events) change the spatial outcome of the scenario being studied. This enables “what-if” planning scenarios that can be visually examined and interpreted for each simulation exercise. A critical component to using the LEAM model is calibration of the weights for the individual model drivers.

Objective

The objective of this project was to create a methodology and tool set to calibrate driver weights for LEAM that (1) uses readily available land-use data, (2) re-quires minimal data preparation, (3) is easily implemented, and (4) is adaptable for various implementations of LEAM. The methodology must provide optimum driver weight calibrations based on available calibration data.

Approach

LEAM is currently calibrated using subject matter expert opinion. An alternate calibration approach was identified as a critical component of the software development process. Calibration methods used in a prior modeling effort were identified as having application in the LEAM system (Fang et al. 2002). The calibration methods were modified for the LEAM system and compared against historic data to verify that the methods were applicable. Validation results are documented in Fang et al. (2005). Once the approach was validated with historic data, the methods were automated. The resulting methodology and tool set is described in detail in this report and demonstrated with data for Fort Benning, GA. All instructions, software code, and scripts are included in appendices that allow users to conduct the analysis for their areas of interest and modify as required for future versions of LEAM.

Scope

Future training and testing capacities on most installations will be directly and indirectly affected by regional land-use policies and investments. Many of the encroachment factors affecting training are a direct result of the surrounding regional urban patterns. These patterns are the result of free-market property exchanges and ownership. Land attractiveness to free market participants (e.g., home owners) is based on local county and city investments and policies including roads, utilities, and zoning. Future training and testing capacities are, in part, a function of these regional land-use investments and policies. LEAM is a tool set developed to help installations assess land-use policies and is applicable to all installations that have sufficient data to populate the model. The effort described in this technical report provides model calibration tools to support implementation of LEAM. The efforts described in this report are applicable to all site-specific implementations of LEAM. Specific LEAM calibrations shown in this report are for demonstration purposes only.

Mode of Technology Transfer

This document and accompanying scripts, code, and digital maps provides part of the technology transfer of the project to military installations, their supporting organizations, and the Office of Economic Adjustment.

This report will be made accessible through the World Wide Web (WWW) at URL:

http://www.cecer.army.mil

2 Overview of Procedures for Estimation of the Weights of LEAM

LEAM predicts urban sprawl and its environmental impact. It is a cellular automata model. One of the core submodels of LEAM uses physical (natural and constructional) and social factors to simulate the dynamics of urban development (Urban Sprawl Submodel). Previous studies developed methods to estimate the weights (coefficients) of the factors using logistic regression (see Appendix A). Currently, more than 13 possible factors are incorporated into this submodel. Due to the amount of available information for a specific region, however, not all the possible factors may be available for modeling. This situation requires that the procedure to estimate the weights of the factors be flexible. PEWL uses a general logistic regression solution and characteristics of the Urban Sprawl Submodel. The PEWL procedure uses computer programs to prepare data samples, generate program scripts for regression analysis, and produce data files (estimated weights) for simulation using the Urban Sprawl Submodel. The generated analysis programs from PEWL are executed using SAS, a very common statistical package, to complete logistic estimation.

PEWL was developed using the FORTRAN language in Microsoft® Developer Studio Fortran PowerStation 4.0, 1993-1994) software. PEWL includes both the source and executable files of the FORTRAN programs.

3 General Settings

PEWL requires a general computer directory (folder) and a set of subdirectories (subfolders). The suggested name of the directory is “LEAM_Calb”. The executable files (and their FORTRAN source files) of PEWL should be copied into this folder. The guide file “Parameters.dat”, which contains the essential information for the estimation procedure, should also be stored in the LEAM_Calb folder. When a PEWL user compiles the FORTRAN source files, they need to make sure that the generated executable files are in this folder and not the “debug” subfolder, which may be automatically generated by some FORTRAN compilers.

At least three subfolders must be created under the “LEAM_Calb” directory. Subfolder “DataFiles” is for original input data files, which include digital maps of land use/cover and model factors (called attractors in LEAM documentation). These digital maps must be in ASCII* format. A subfolder named “Sampled” is for storage of the data files that contain the random samples for estimation. The subfolder “Estimatn” is for SAS programs and their output files.

PEWL contains a guide file and three executable (and FORTRAN source) files. Table 1 lists the steps from setting up PEWL to generating the final product, the estimated weights of LEAM. The details of each step are described in subsequent chapters.

* ASCII = American Standard Code for Information Interchange

Table 1. PEWL set up.

Step	Task	Product/Location	System
1	Set up PEWL in computers	Copied files in created folders	Operation
2	Prepare digital map files	Maps in Subfolder “DataFiles”	GIS
3	Create the guide file	File “Parameters.dat” in Folder “LEAM_Calb”	Notepad
4	Draw random samples (run “LEAM_Sampling.exe”)	Data files in Subfolder “Sampled”	PEWL
5	Generate SAS programs (run “LEAM_SAS_Builder.exe”)	SAS programs in Subfolder “Estimatn”	PEWL
6	Estimate the weights (run SAS programs generated at Step 5)	Estimates on SAS Output screen	SAS
7	Create data files of estimates	Data files of estimates in Subfolder “Esti-matn”	Notepad
8	Integrate the estimates (run “LEAM_Est_Integratn.exe”)	Data file “Est_All.dat” in Subfolder “Estimatn”	PEWL

4 Guide File

PEWL uses a guide file named “Parameters.dat”. The guide file contains all essential parameters for drawing random samples, generating SAS programs, and integrating the output files of SAS programs. Figure 1 shows the format of this file.

The first line of the guide file is the title line, which is a brief description of the purpose of the guide file. The guide file has seven sections, each of which starts with an indication line (information line). Information lines provide the in-formation for users to input correct parameters in the corresponding sections. Both title and information lines can be modified according to the user’s preference. These lines cannot be removed, however, and each information or title line should occupy exactly one row. No space line (row) is allowed between any two lines (rows) of title/ information and subsequent parameters.

The guide file contains parameters for the folder location, the dimension of all maps, and the file names of land-use and factor maps. The first parameter section provides the location of the general folder “LEAM_Calb.” The location should be a drive or folder (directory) named with fewer than 60 characters. When inputting the location of the general folder, it is important to key in the “\” sign as the last character, to indicate a folder name. For example, when the location of the general folder “LEAM_Calb” is “D:\test\” and the directory of “LEAM_Calb” is “D:\test\LEAM_Calb\”, the correct location parameter in the guide file should be “D:\test”, instead of “D:\test\”.

Section 2 of the guide file provides the dimension of all digital maps. The first dimension parameter is the number of pixel rows of digital maps. The second parameter is the number of pixel columns of the maps. The dimensions of all maps have to be the same before generating the guide file.

Section 3 provides the file names of land-use/cover maps. Two historical maps must be provided in chronological order. The first map represents the oldest period of time.

The next four sections of the guide file provide the information about the factors used in logistic modeling. In the example (see Figure 1), a “factor” is called an “attractor” in the Information Lines. First, the number of factors needs to be given

(Section 4). Then the file names of the factor maps are listed by the classes of development. Sections 5, 6, and 7, respectively, list the file names for modeling the probability of possible land use converted to Residential, Commercial-Industrial, and Open Space. The order of the classes of development should not be changed. In each of these sections, under the Information Line, the number of file names in the list has to be the same as given in Section 3. As long as a factor map is used for modeling the probability of one kind of land-use conversion, the file name of that factor has to be included in the file list of that kind of conversion, no matter if the file name of that factor has or has not been listed in the sections for other kinds of land-use conversion. Therefore, if a factor has the same effect on all three kinds of land-use conversion, the same file name of that factor has to be included in all file list sections. Users can determine the order of the file name lists according to their preferences. The order must be the same for all file name lists, however, since PEWL uses the order of the file name lists to sort/arrange the factors. Thus, varying the order of files in different file lists will cause modeling errors.

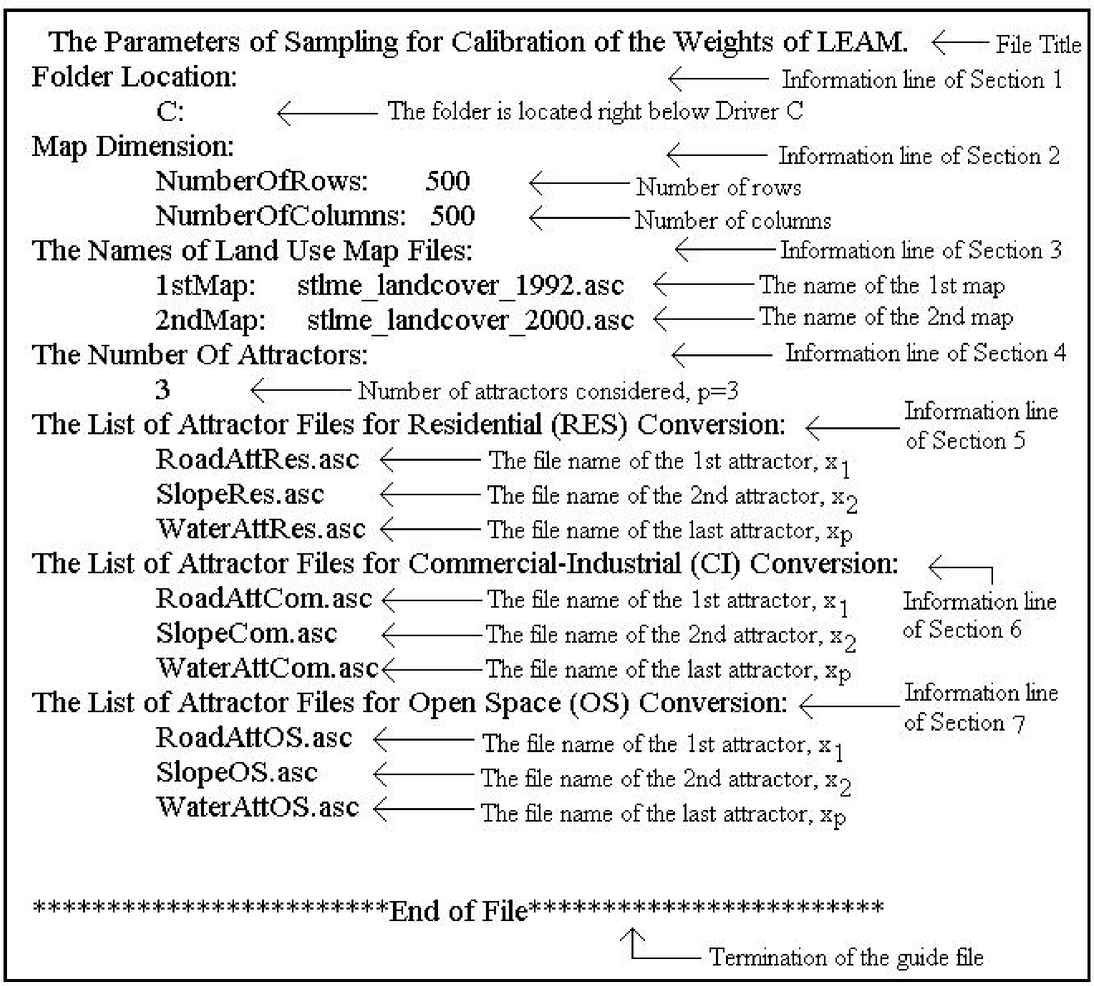

Figure 1. Format of the guide file (example).

5 Sampling

Based on parameters provided in the guide file, random samples will be generated by running the program "LEAM_Sampling.exe" (or by first compiling "LEAM_Sampling.f" to obtain the executable file). Before running this program, all digital maps named in the guide file have to be prepared and stored in the subfolder (subdirectory) "DataFiles."

Sample size is automatically determined by the PEWL program. When the total number of pixels in a digital map is less than 350,000, about 2.5 percent of pixels will be randomly sampled. Otherwise, about 9,500 pixels will be randomly sampled for estimation. The random samples are saved in four data files in the subfolder "Sampled." The names of the data files are "RES_Sampled.dat," "CI_Sampled.dat," "OS_Sampled.dat," and "Nch_Sampled.dat," which are used respectively for estimating the weights (coefficients) of probability models for converting to Residential, Commercial-Industrial, and Open Space use, and to "No development."

Each of the four data files has the same general structure. In a data file, the first column is the code of land-use conversion. When a sampled pixel changed from available to a specific land use (Residential, Commercial-Industrial, or Open Space) as indicated by the file name, the code is 0.0000; otherwise, the code is 1.0000. This code system is based on the definitions of SAS logistic procedure, which uses 0 (zero) to represent the positive response (happened) of an event. From the second column to the last (p+1) column, there are totally p (the number of factors used for modeling as defined in Section 3 of the guide file) columns. The order of these columns corresponds to those defined in the data file list in the guide file. For example, in the sampled data file "CI_Sampled.dat," the first column is the code to indicate the conversion from other land uses to Commercial-Industrial use. If the code of an observation has the value of 0.0000, it indicates that that pixel has been converted from other land use to Commercial-Industrial use. When the code has a value of 1.0000, it indicates that that pixel either had not been changed or had been changed to Residential or Open Space use. According to Section 5 of the guide file, the order of the digital factor maps is "RoadAttCom.asc," "SlopeCom.asc," and "WaterAttCom.asc." Therefore, from the second to fourth (1 + p, p=3) columns, the values are from the factors represented by "RoadAttCom," "SlopeCom," and "WaterAttCom," respectively. Figure 2 illustrates the structure of the sampled data files.

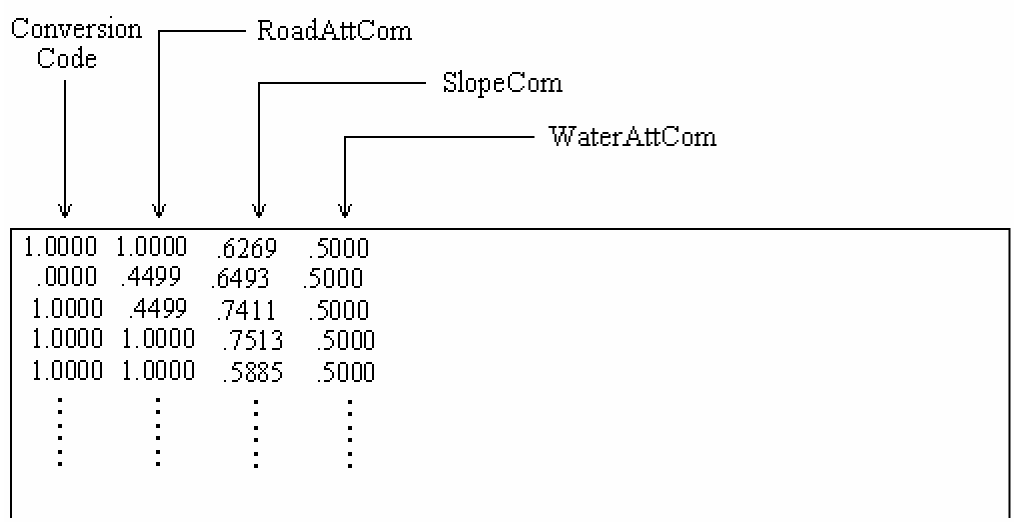

Figure 2. Structure of sampled data files; illustrated file is “CI_Sampled.dat”.

6 SAS Program Builder

SAS programs are needed to estimate the weights (coefficients) of LEAM. The required SAS programs are generated by running “LEAM_SAS_Builder.exe” (or first compiling “LEAM_SAS_Builder.f” to obtain the executable file). The generated SAS programs are stored in subfolder “Estimatn”.

Four SAS programs are generated by running the SAS program builder. For each land-use conversion, one SAS program is generated for estimating the weights of the corresponding probability model. For the probability of converting to land use for Residential, Commercial-Industrial, and Open Space, the names of the correspond-ing SAS programs are respectively “RES_Est.sas”, “CI_Est.sas”, and “OS_Est.sas”. SAS program “NCH_Est.sas” is for estimating the weights of the model to predict the probability of “no conversion”. In each of these SAS programs, the independent variables of the logistic models are the factors and their cross products (to represent the factor interactions). The factors are named as variables, and their cross products are named as variables (m=[p-1]×p/2). The order of the corresponding map file list (Sections 5–7 of the guide file) is the order of the independent variables X’s, and is printed at the beginning of the SAS program as message lines. Independent variables (U’s) are automatically generated and defined in the SAS programs.

Since no factor is defined for modeling “no conversion,” the independent variables X’s of this model are created using the independent variable X’s of the models for the three types of conversion. Independent variable U’s of the model for “no conversion” are generated and defined in the SAS programs in a manner similar to the variables for land-use conversion described in the preceding paragraph.

7 Integration of Estimates

The weights (coefficients) of LEAM are estimated by using the SAS software program. The generated SAS programs "RES_Est.sas", "CI_Est.sas", "OS_Est.sas", and "NCH_Est.sas" estimate the weights for corresponding land-use changes with specific sampled data files (Table 2). You will need to manually generates data files that contain the estimates of the weights (coefficients). Table 2 lists the names of those estimate files.

After running one of the SAS programs listed in Table 2, the outcome of logistic regression will be displayed on the SAS Output Screen (see Figure 3 for an example). Highlight the estimates part emphasized in Figure 3, copy-paste that part to Notepad, add a Title Line and a space line at the top of the context, then save it as a text file with the name according to the SAS program and Table 2. Figure 4 illustrates the format of the files of estimates.

When all files of estimates listed in Table 2 have been developed, the estimates of all weights are integrated into one file, "Est_All.dat". The executable file "LEAM_Est_Integratn.exe" (which can also be obtained by compiling "LEAM_Est_Integratn.f") integrates the estimates of the weights of LEAM from the four independent files of estimates to create "Est_All.dat". This integrated file of estimates is the input source of weights of LEAM for simulation using programs in C/C++, FORTRAN, or other computer languages. Figure 5 illustrates the format of "Est_All.dat."

Table 2. Types of land-use conversion and the names of corresponding SAS programs, sampled data files, and estimated weights files.

Land-use Conversion	SAS Program	Sampled Data File	File of Estimates
Residential (RES)	RES_Est.sas	RES_Sampled.txt	Est_RES.txt
Commercial-Industrial (CI)	CI_Est.sas	CI_Sampled.txt	Est_CI.txt
Open Space (OS)	OS_Est.sas	OS_Sampled.txt	Est_OS.txt
No Change (NCH)	NCH_Est.sas	Nch_Sampled.txt	Est_NCH.txt

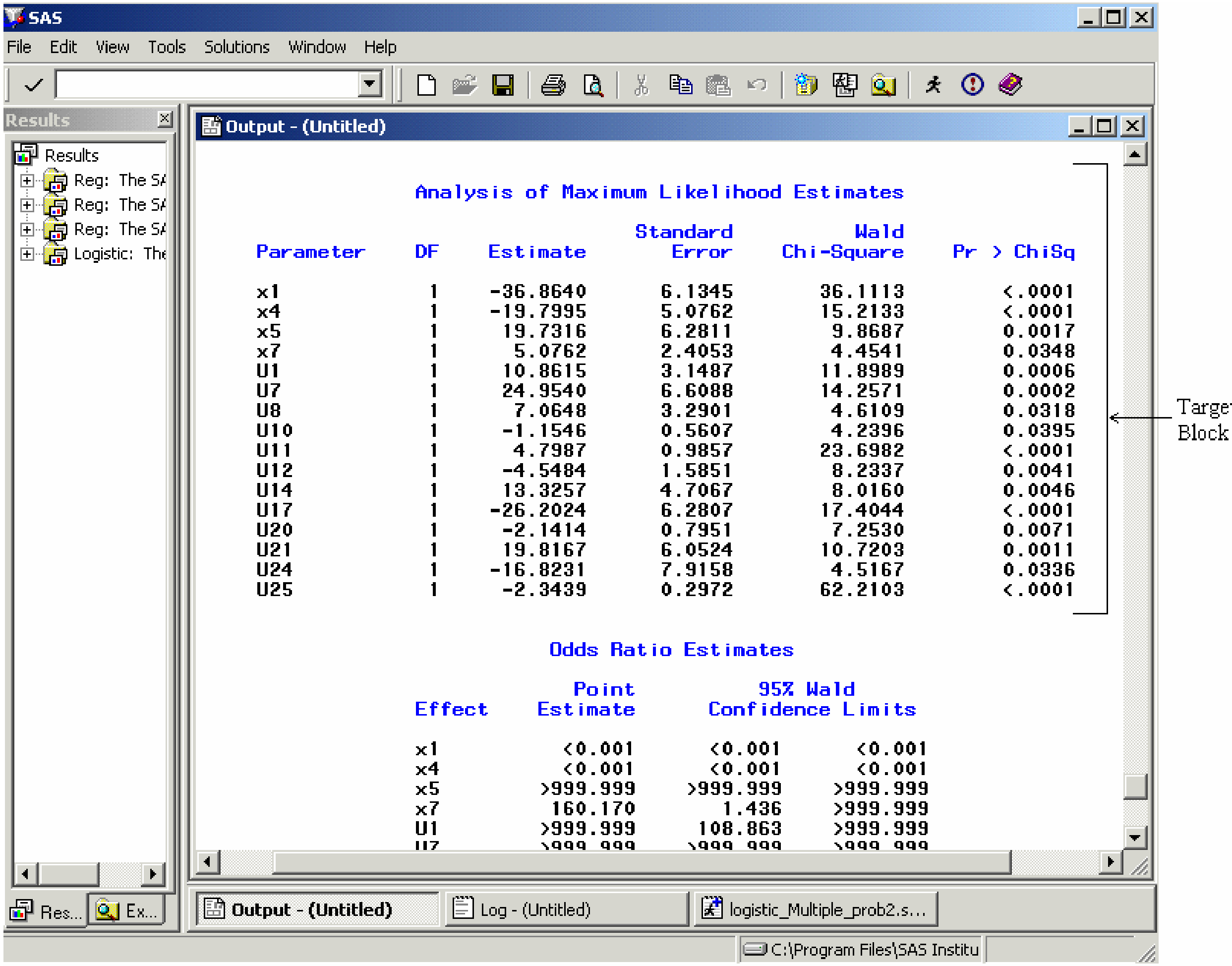

Figure 3. SAS Output Screen displaying estimates of weights of LEAM (for Residential, eight factors are considered). The part labeled "Target Block," which is at almost the end of the Output Screen, is what needs to be highlighted, copied, and pasted into Notepad to create a file of estimates.

Weights of LEAM for prediction of the probability of No change ← Title line

← Space line

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
x2	1	19.5595	3.9388	24.6593	<.0001
x3	1	9.8297	3.4374	8.1774	0.0042
x5	1	-34.6420	4.1868	68.4593	<.0001
x6	1	8.6733	1.5526	31.2056	<.0001
x8	1	15.4132	4.1809	13.5908	0.0002
u2	1	2.6182	1.1498	5.1850	0.0228
u3	1	3.7806	1.7510	4.6619	0.0308
u4	1	12.4261	2.0478	36.8205	<.0001
u7	1	5.2639	1.8742	7.8880	0.0050
u8	1	-3.4480	1.1476	9.0273	0.0027
u11	1	-4.4063	1.5246	8.3534	0.0038
u14	1	-2.8152	1.3525	4.3329	0.0374
u20	1	3.7414	1.7683	4.4767	0.0344
u26	1	-15.4770	3.2647	22.4745	<.0001

← Pasted from SAS Output Screen

Figure 4. Format of the files of estimates. The file of estimates in this example is "Est_NCH.txt," which is for "no change" (eight factors are considered).

Estimates of the weights of LEAM ← Title line

← Space line

Attractor	RES	Com_Ind	Open Space	No_Change
X1	-.368640E+02	-.268640E+02	-.373906E+02	.000000E+00
X2	.000000E+00	-.157995E+02	.195595E+02	.195595E+02
X3	.000000E+00	.227316E+02	.982970E+01	.982970E+01
X4	-.197995E+02	.000000E+00	.000000E+00	.000000E+00
X5	.197316E+02	.000000E+00	-.346420E+02	-.346420E+02
X6	.000000E+00	.607620E+01	.000000E+00	.867330E+01
X7	.507620E+01	.000000E+00	.867330E+01	.000000E+00
X8	.000000E+00	.000000E+00	.154132E+02	.154132E+02
U1	.108615E+02	.118615E+02	.000000E+00	.000000E+00
⋮	⋮	⋮	⋮	⋮
U26	.000000E+00	.000000E+00	.000000E+00	-.154770E+02
U27	.000000E+00	-.234390E+01	.000000E+00	.000000E+00
U28	.000000E+00	.000000E+00	.000000E+00	.000000E+00

← Space line

← Space line

Figure 5. Format of the integrated estimates (file "Est_All.dat"). In this example, eight factors are considered in LEAM.

8 Conclusions

This report documents methodology and a tool set to calibrate the driver weights for LEAM that (1) uses readily available land-use data, (2) requires minimal data preparation, (3) is easily implemented, and (4) is adaptable for various implementations of LEAM. Implementing this methodology using the tools documented in this report provides optimum driver weight calibrations based on the available calibration data.

References

Deal, Brian. (2001). The Military Land use Evolution and impact Assessment Model (LEAM), Engineer Research and Development Center (ERDC), Construction Engineering Research Laboratory (CERL), Champaign, IL, Technical Note (TN)-01-2, ADA408969, November 2001.

Deal, Brian M., Diane M. Timlin, and William D. Goran. (2002). *Urban Encroachment of Military Lands: Military Installations at Risk*, ERDC/CERL Technical Report (TR)-02-3, ADA401147, January 2002.

Fang, S., S. Wente, G.Z. Gertner, G. Wang, and A.B. Anderson. (2002). "Uncertainty analysis of predicted disturbance from off-road vehicular traffic in complex landscapes at Fort Hood." *Environmental Management*, 30(2):199-208.

Fang, S., G.Z. Gertner, Z. Sun, and A.B. Anderson. (2005). "The Impact of interactions in spatial simulation of the dynamics of urban sprawl." *Landscape and Urban Planning*, 72(4):294-306.

Goran, W.D., B. Deal, T. Aden, D. Timlin. (2000). "Urban Growth: A Challenge to Sustaining U.S. Department of Defense Land Uses," in Brunckhorst, D.J. (ed)., *Bioregional Planning: Resources Management Beyond the New Millenium*. Harwood Academic: Amsterdam. ISBN 90-5823-046-5.

Jenicek, Elisabeth M., Brian M. Deal, and Adam Sagert. (2002). *Sustainability Indices for Military Installations: An Annotated Bibliography*, ERDC/CERL TR-02-25, ADA407531, September 2002.

Lozar, Robert C., C.R. Ehlschlaeger, and Jennifer Cox (2003). *A Geographic Information Systems (GIS) and Imagery Approach to Historical Urban Growth Trends Around Military Installations.* ERDC/CERL TR-03-9, ADM001569, May 2003.

Orland, B., K. Budthimedhee, and J. Uusitalo. (2001). "Considering Virtual Worlds as Representations of Landscape Realities and as Tools for Landscape Planning," *Landscape and Urban Planning*. Vol 54, pp 139-148, 25 May 2001.

Appendix A: Estimation of the Weights (Coefficients) of LEAM

Based on an earlier study, a logistic model including both individual attractors and their interaction (product terms) was used to predict the probability of Residential development:

Equation A1. Initial model for predicting the probability of residential development.

$$\mathrm{Log}\left(\frac{P(\mathrm{C_k}\,|_{Non->\mathrm{C_k}})}{1-P(\mathrm{C_k}\,|_{Non->\mathrm{C_k}})}\right)=\sum_{k=1}^{13}\mathrm{A_i x_i}+\sum_{j=1}^{36}\mathrm{B_j U_j}\ ,\ \mathrm{k}=1,2,3,4;$$

where X’s and U’s are the attractors of LEAM, their combinations listed in Table A1, and A’s and B’s are weights (coefficients) estimated based on the historical maps. $P(\mathrm{C_k}\,\mathrm{Res}|Non->\mathrm{C_k})$ is the probability of the k^{th} category of development. When some of the factors listed in Table A1 are missing in a specific study area, the corresponding terms in Eq. A1 will be dropped in model calibration as well as in probability prediction.

Multinomial probabilities were predicted in two steps. First, a set of multinomial probability models had been built using selective logistic regression analysis. Then, a standardization procedure was applied to compute the multinomial probabilities of the categories. The following equations are provided for each probability estimate.

Equation A2. Initial model for predicting the probability of Commercial-Industrial.

$$\mathrm{Log}\left(\frac{\mathrm{P'_{CI}}}{1-\mathrm{P'_{CI}}}\right)=\sum_{i=1}^{13}\mathrm{A\_CI_i \times X\_CI_i}+\sum_{j=1}^{36}\mathrm{B\_CI_j \times U\_CI_j}$$

Equation A3. Initial model for predicting the probability of Open Space.

$$\mathrm{Log}\left(\frac{\mathrm{P'_{OS}}}{1-\mathrm{P'_{OS}}}\right)=\sum_{i=1}^{13}\mathrm{A\_OS_i \times X\_OS_i}+\sum_{j=1}^{36}\mathrm{B\_OS_j \times U\_OS_j}$$

Equation A4. Initial model for predicting the probability of Residential.

$$\mathrm{Log}\left(\frac{P'_{RE}}{1-P'_{RE}}\right)=\sum_{i=1}^{13} A\_RE_i \times X\_RE_i + \sum_{j=1}^{36} B\_RE_j \times U\_RE_j$$

Equation A5. Initial model for predicting the probability of No Change.

$$\mathrm{Log}\left(\frac{P_{NC}}{1-P_{NC}}\right)=\sum_{i=1}^{13} A\_NC_i \times X\_NC_i + \sum_{j=1}^{36} B\_NC_j \times U\_NC_j$$

where A_Ci is the coefficient of the score of attractor i, where C={RE, CI, OS, NC}={Residential, Commercial/industrial, Open Space, No Change}, X_W_i is the score of attractor i, $B\_\_C_j$ is the coefficient of the product of the scores of a pair of attractors (listed in Table A1), $U\_C_j=C_m * C_n$, and $NC_k=(3-X\_RE_k-X\_CI_k-X\_OS_k)/3$, k=1, …, 8.

To make the predicted probabilities consistent, the original probabilities predicted using the logistic models from Eqs. A2–A5 need to be adjusted:

Equation A6. Probability adjustment for initial model probabilities.

$$P_{LUC} = \frac{P'_{LUC}}{P'_{CI} + P'_{OS} + P'_{RE} + P'_{NC}}$$

where LUC={RE, CI, OS, NC}={land use converted to Residential, Commercial/industrial, Open Space, No Change}.

Table A1. Codes of factors and their combinations in Equation A1.

Factor	*Code*	Factor interaction	*Code*	Factor interaction	*Code*
City Attractor	X1	**X1*X2**	U1	X4*X5	U19
City Road Attractor	X2	X1*X3	U2	X4*X6	U20
Forest Attractor	X3	X1*X4	U3	X4*X7	U21
Slope Attractor	X4	X1*X5	U4	X4*X8	U22
Ramp Attractor	X5	X1*X6	U5	X5*X6	U23
Road Intersection	X6	X1*X7	U6	X5*X7	U24
State Highway	X7	X1*X8	U7	X5*X8	U25
Water Attractor	X8	X2*X3	U8	X6*X7	U26
Neighbor	X9	X2*X4	U9	X6*X8	U27
Utilities	X10	X2*X5	U10	X7*X8	U28
Growth Trend	X11	X2*X6	U11	X9*X1	U29
Agriculture Protection	X12	X2*X7	U12	X9*X2	U30
Growth Booster	X13	X2*X8	U13	X9*X3	U31
~~	~~	X3*X4	U14	X9*X4	U32
		X3*X5	U15	X9*X5	U33
		X3*X6	U16	X9*X6	U34
		X3*X7	U17	X9*X7	U35
		X3*X8	U18	X9*X8	U36

Nb_nc = 0.33*(3-nbR - nbC - nbO),
where {nbR, nbC, nbO} = {neighbor Residential, neighbor Com.-Ind., neighbor Open_Space}.
Ut_nc = 0.33*(3 - Ut_R - Ut_C - Ut_O).

Appendix B: PEWL Fort Benning Example

In this example, the land-use conversion from 1980 through 1990 inside and around Columbus City, GA, Phenix City, AL, and Fort Benning are considered. The land-use maps from 1980 and 1990 and the maps of the considered factors (attractors) are included in subfolder "Benning_Example" inside the folder "LEAM_Calb". The corresponding guide file "Parameters.dat", which lists all the essential parameters and file lists for weight estimation, is also included in subfolder "Benning_Example". In order to run the example, all map files need to be moved into subfolder "DataFiles", and the guide file "Parameters.dat" needs to be moved to folder "LEAM_Calb". Then, follow the instruction in this User's Guide to estimate the weights of LEAM.

Appendix C: FORTRAN Source Code

LEAM_Sampling.f Program Code

```
*****************************************************************************
c           Programmer:  Shoufan Fang
c
c     Program to sample obs. for estimating the weights of LEAM
c
      program   Sampler
c
      USE MSIMSL

      character chtmp*100, FName*50, filename*40,dirc*85
      integer nrows, ncols, N_S, N_files, L_Dirc

      CALL DISCLM()

      open(unit=30,file="Parameters.dat", form='formatted')

      read(30,*)
      read(30,*)
c   Read Directory
      read(30,*) dirc

      do i=1,85
        if (dirc(i:i) .NE. '')  L_Dirc=i
      enddo

      read(30,*)
      read(30,*) chtmp,nrows
      read(30,*) chtmp,ncols
c
      read(30,*)
      read(30,*) chtmp,filename
      FName='DataFiles\'//filename
      chtmp=dirc(1:L_Dirc)//'\LEAM_Calb\'//FName
      open(unit=1,file=chtmp, form='formatted')
      read(30,*) chtmp,filename
      FName='DataFiles\'//filename
      chtmp=dirc(1:L_Dirc)//'\LEAM_Calb\'//FName
      open(unit=2,file=chtmp, form='formatted')
```

```
c
      chtmp=dirc(1:L_Dirc)//'\LEAM_Calb\DataFiles\Class_Coords.dat'
      open(unit=15,file=chtmp)
      N_S=0
      CALL Sampl_LU(nrows, ncols,N_S)
CCCC********CCCCCC*******CCCCCCCC********
      read(30,*)
      read(30,*) N_files
c
      chtmp=dirc(1:L_Dirc)//'\LEAM_Calb\DataFiles\Class_Coords.dat'
      open(unit=25,file=chtmp, form='formatted')
c
      CALL SamplATTR(nrows,ncols,N_S,N_files,dirc)



ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c End of program
350   stop
      end

cccccc

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c   Subroutine to determine which pixels should be sampled and sample land use
conversion
      Subroutine    Sampl_LU(nrs, ncs,NS)
c     character folder*85
      integer nrs, ncs, NS, n(18)
      real x(8,nrs, ncs),a(ncs),rndm
      real*8 acpt1,acpt2

      do nn=1,2
         do j=1,6
           read(nn,*)
         enddo
c
         do i=1,nrs
           read(nn,*)(x(nn,i,j),j=1,ncs)
c     print*,nn,i,nrows,ncols
         enddo
         close (nn)
      enddo

      a(1)=1.0*nrs*ncs
      if (a(1) .LT. 350000) then
         acpt1=0.025
      else
         acpt1=9500.0/(1.0*nrs*ncs)
      endif
```

```
      acpt2=0.5+acpt1
      acpt1=0.5

      NS=0
       do i=3,nrs-2
         do j=3,ncs-2

            a(j)=FUNC_Possbl_LU(x(1,i,j))
            if (a(j) .EQ. 0) goto 580
            if (x(1,i,j) .LE. -999) goto 580

            a(j)= x(1,i,j)
            n(1)= int(FUNC_Dvlp_LU(a(j)))
            a(j)= x(2,i,j)
            n(2)= int(FUNC_Dvlp_LU(a(j)))
c
            n(8)=-10
c
c     Changed?
            n(3) = n(2)-n(1)
            if (n(3) .NE. n(2)) n(3)=0
c
            rndm=RNUNF()
            if ((rndm .GE. acpt1) .AND. (rndm .LE. acpt2)) then
               NS=NS+1
ccccccccccc  Initial land use
               n(5)=int(x(1,i,j))
               write(15,201)n(3),n(5),i,j
            endif
c           read(nn,*)(x(nn,i,j),j=1,ncols)
580      enddo
       enddo
       close (15)
       print*,'Number of pixels sampled: ',NS
       print*,' '
       print*,' '

201    format(5(1x,I5))

      return
      end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c   Sampling the factor maps according to the determined coordinates
      Subroutine SamplATTR(nrs,ncs,NS,N_f,folder)

      character FName*45, filename*30,nm_f(3)*15,folder*85,chtmp*100
      integer nrs, ncs, n_cd(4,NS),NS, NN_S
      real x(N_f,nrs,ncs)
      data nm_f /'RES_sampled.dat','CI_sampled.dat','OS_sampled.dat'/
```

```
      do i1=1,NS
         read(25,*)(n_cd(k,i1),k=1,4)
      enddo
         close (25)

      do i1=1,85
         if (folder(i1:i1) .NE. '') NN_S=i1
      enddo


      do II=1,3
       NN=II+35
       FName='Sampled\'//nm_f(II)
       chtmp=folder(1:NN_S)//'\LEAM_Calb\'//FName
       open(unit=NN,file=chtmp)
       read(30,*)

       do LL=1,N_f
           read(30,*) filename
           FName='DataFiles\'//filename
           chtmp=folder(1:NN_S)//'\LEAM_Calb\'//FName
           open(unit=LL,file=chtmp, form='formatted')

           do i=1,6
              read(LL,*)
           enddo

           do i=1,nrs
              read(LL,*)(x(LL,i,j),j=1,ncs)
           enddo
           close (LL)
       enddo


       do i=1,NS
          xy=1.0
          if (n_cd(1,i) .EQ. II) xy=0.0
          write(NN,110)xy,(x(k,n_cd(3,i),n_cd(4,i)),k=1,N_f)
       enddo
       close (NN)

      enddo
cccccccccccccccccccccccccccc
c
      do II=1,3
       FName='Sampled\'//nm_f(II)
       chtmp=folder(1:NN_S)//'\LEAM_Calb\'//FName
       open(unit=II,file=chtmp, form='formatted')
      enddo
```

```
      chtmp=folder(1:NN_S)//'\LEAM_Calb\Sampled\Nch_Sampled.dat'
      open(unit=15,file=chtmp)

      do i=1,NS
        xy=1.0
        if (n_cd(1,i) .EQ. 0) xy=0.0
        do II=1,3
          read(II,*)xz,(x(k,II,1),k=1,N_f)
        enddo
        Do LK=1, N_f
          x(LK,1,5)=(3.0-x(LK,1,1)-x(LK,2,1)-x(LK,3,1))/3.0
        enddo
        write(15,110)xy,(x(k,1,5),k=1,N_f)
      enddo

        do II=1,3
          close (II)
        enddo
        close (15)

cccccccccccccccccccccccccccccc
110   format(17(1x,f6.4))

       return
       end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
c  Information subroutine
      Subroutine DISCLM()

      print*,'*******************************************************'
      print*,'*                                                     *'
      print*,'*                    DISCLAIM                         *'
      print*,'*                                                     *'
      print*,'*                                                     *'
      print*,'*      The authors of this program are not responsible  *'
      print*,'*  to any computer hardware damage or information loss  *'
      print*,'*  due to improper use of this program.                 *'
      print*,'*                                                     *'
      print*,'*                                                     *'
      print*,'*                                                     *'
      print*,'*       ALL RIGHTS OF THIS PROGRAM ARE RESERVED         *'
      print*,'*                                                     *'
      print*,'*                                                     *'
      print*,'*******************************************************'
      print*,' '
      print*,' '
c
       pause
```

```
c
     do LK=1,10
       print*,' '
     enddo
c
     print*,'**********************************************************'
     print*,'                                              '
     print*,'                                              '
     print*,'    Sampling for estimation of the weights of LEAM     '
     print*,'                                              '
     print*,'                                              '
     print*,'**********************************************************'
     print*,' '
     do LK=1,5
       print*,' '
     enddo

     return
     end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      function FUNC_WATER(w)

      real w, q

      q = 0
      IF ((w .EQ. 11) .or. (w .EQ. 12)) q = 1
      FUNC_WATER=q
      return
      end
ccc******+++++-----=====
c The folowings are working functions according to LEAM
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      function FUNC_CMI(w)

      real w, q

      q = 0
      IF ((w .GE. 23) .and. (w .LE. 24)) q = 1
      FUNC_CMI=q
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      function FUNC_OPN(w)

      real w, q

      q = 0
      IF  (w .EQ. 85)  q = 1
      FUNC_OPN=q
      return
```

```
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      function FUNC_RES(w)

      real w, q

      q = 0
      IF ((w .GE. 21) .and. (w .LE. 22)) q = 1
      FUNC_RES=q
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      function FUNC_Dvlp_LU(w)

      real w, q

      q = 0
      IF ((w .EQ. 21) .or. (w .EQ. 22)) q = 1
      IF ((w .EQ. 23) .or. (w .EQ. 24)) q = 2
      IF (w .EQ. 85) q = 3

      FUNC_Dvlp_LU=q

      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      function FUNC_Possbl_LU(w)

      real w, q

      q = 1
      IF ((w .EQ. 21) .or. (w .EQ. 22)) q = 0
      IF ((w .EQ. 23) .or. (w .EQ. 24)) q = 0
      IF ((w .EQ. 11) .or. (w .EQ. 85)) q = 0

      FUNC_Possbl_LU=q
      return
      end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCC

****************************************************************************
```

LEAM_SAS_Builder.f Program Code

```
***************************************************************************
 c          Programmer:  Shoufan Fang
c
c    Program to generate SAS programs for estimating the weights of LEAM model
c
c
c

     program   SAS_Builder
c
c
     Character folder*85
     Integer N_files

     CALL DISCLM()

     open(unit=30,file="Parameters.dat", form='formatted')


     Do i=1, 10
        if (i .EQ. 3) then
           read(30,*) folder
        else
           read(30,*)
        endif
     Enddo

     read(30,*) N_files
     read(30,*)
c
     CALL SAS_PROGRM(N_files,folder)
c
      close (30)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c End of program
350     stop
     end

cccccc

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
     Subroutine     SAS_INTI(M)
     Integer M

     Write (M,*) 'options ps=1000 ls=90;'
     Write (M,*)

     return
```

```
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      Subroutine     SAS_PROGRM(N_F,fldr)
      integer N_F, N_VAR, nn(10), N_FL
      character chtmp*125, VName(N_F)*50, SAS_NM(4)*12,cr(2)*1
      character F_nm(4)*15,DtNm(4)*5,resp(4)*3,cc(5)*3 ,fldr*85
c
      data F_nm/'RES_Sampled.dat','CI_Sampled.dat','OS_Sampled.dat',
     +          'NCh_Sampled.dat'/
      data DtNm/'one','two','three','four'/
      data resp/'RES','CI','OS','NCh'/
      data SAS_NM/'RES_Est.sas','CI_Est.sas','OS_Est.sas','NCh_Est.sas'/
c
      do i=1,85
         if (fldr(i:i) .NE. '') N_FL=i
      enddo

      do i=1,4
         chtmp=fldr(1:N_FL)//"\LEAM_Calb\Estimatn\"//SAS_NM(i)
         open(unit=i,file=chtmp)
         CALL SAS_INTI(i)
      enddo

      cr(1)=CHAR(39)
      cr(2)=CHAR(59)

       do LL=1,N_F
          CALL N_CHAR(LL, cc(1))
          read(30,*) VName(LL)
          do JK=1,len(VName(LL))
             if (VName(LL)(JK:JK) .NE. '') nn(10)=JK
          enddo
          nn(10)=nn(10)+1
          chtmp="* "//" x"//cc(1)//"= "//VName(LL)(1:nn(10))//cr(2)
          do i=1,4
                Write (i,*) chtmp
          enddo
       enddo
       do i=1,4
           Write (i,*)
           Write (i,*)
       enddo

c      F_nm='_Sample.dat'
      N_VAR=N_F*(N_F-1)/2

       CALL N_CHAR(N_F, cc(5))

       CALL N_CHAR(N_VAR, cc(4))
```

```
        do JJ=1,4
           chtmp="Data "//DtNm(JJ)//cr(2)
            Write (JJ,*) chtmp
            CALL GETFILE(F_nm(JJ),resp(JJ),cc(5),JJ,fldr)
            nn(JJ)=0
            do II=1,N_F -1
               do IJ = II+1, N_F
                  nn(JJ)=nn(JJ)+1
                  CALL N_CHAR(nn(JJ), cc(1))
                  CALL N_CHAR(II, cc(2))
                  CALL N_CHAR(IJ, cc(3))
            chtmp="U"//cc(1)//"= x"//cc(2)//" * x"//cc(3)//cr(2)
                  Write (JJ,*) chtmp
               enddo
            enddo
            Write (JJ,*)
            Write (JJ,*)
c
            CALL SETMODEL(resp(JJ),DtNm(JJ),cc(5),cc(4),JJ)
            close (JJ)
        enddo


c 201    format(5(1x,I5))

      return
      end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      Subroutine GETFILE(Fname,rsp,cnv,M,fold)
      character Fname*15,chr*100,cnv*3,cr(2)*3,rsp*3,fold*85
      Integer M , NFD

      do KL=1,85
         if (fold(KL:KL) .NE. '') NFD=KL
      enddo

      cr(1)=CHAR(39)
      cr(2)=CHAR(59)
      chr="infile "//cr(1)//fold(1:NFD)//"\LEAM_Calb\Sampled\"//
     +    Fname//cr(1)//cr(2)
      Write (M,*) chr
      chr="Input "//rsp//"  x1 - x"//cnv//cr(2)
      Write (M,*) chr
      return
      end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      Subroutine N_CHAR(n5, ccc)
      character  ccc*3, ct(2)*1
      Integer n5, n1, n2
```

```
      n1=int(n5/10)
      n2=n5-10*n1
      ct(1)=CHAR(48+n2)
      ccc=ct(1)
      if (n1 .GT. 0) then
         ct(2)=CHAR(48+n1)
         ccc=ct(2)//ct(1)
      endif

      return
      end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      Subroutine SETMODEL(rsp,DataNm,cvn,ccvn,M)
      character chr*85,DataNm*5,rsp*3,ccr*3,cvn*3,ccvn*3
      Integer M

      ccr=CHAR(59)
      chr="PROC LOGISTIC data="//DataNm//ccr
      Write (M,*) chr
      chr="Model "//rsp//" = x1 - x"//cvn//" U1 - U"//ccvn//
      + " / noint rsq selection=backward "//ccr
      Write (M,*) chr
      chr="RUN"//ccr
      Write (M,*) chr
      Write (M,*)
      return
      end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      Subroutine DISCLM()

      print*,'*********************************************************'
      print*,'*                                        *'
      print*,'*                DISCLAIM                        *'
      print*,'*                                        *'
      print*,'*                                        *'
      print*,'*     The authors of this program are not responsible  *'
      print*,'*  to any computer hardware damage or information loss  *'
      print*,'*  due to improper use of this program.              *'
      print*,'*                                        *'
      print*,'*                                        *'
      print*,'*                                        *'
      print*,'*       ALL RIGHTS OF THIS PROGRAM ARE RESERVED       *'
      print*,'*                                        *'
      print*,'*                                        *'
      print*,'*********************************************************'
      print*,' '
      print*,' '
c
       pause
c
```

```
      do LK=1,10
        print*,' '
      enddo
c
      print*,'*****************************************************'
      print*,'                                                     '
      print*,'                                                     '
      print*,'            Generating SAS Program for               '
      print*,'                                                     '
      print*,'          estimation of the weights of LEAM          '
      print*,'                                                     '
      print*,'*****************************************************'
      print*,' '
      do LK=1,5
        print*,' '
      enddo

      return
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

**************************************************************************
```

LEAM_Est_Integratn.f Program Code

```
**************************************************************************
c          Programmer:  Shoufan Fang
c
c    Program to integrate all estimates of the weights of LEAM
c
c
c

      program   Est_integrator
c
c
      Integer N_files , N_V

      CALL DISCLM()

      open(unit=30,file="Parameters.dat", form='formatted')

      Do i=1, 10
         read(30,*)
      Enddo

      read(30,*) N_files
      read(30,*)
```

```
c
       close (30)
c
      N_V=N_files*(N_files-1)/2

      print*, N_files, N_V

      CALL Integrtn(N_files, N_V)
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c End of program
350     stop
      end

cccccc

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      Subroutine    OutFile()
      character chtmp*125


      open(unit=15,file='Estimatn\Est_All.dat')

      Write (15,*) '         Estimates of the weights of LEAM'
      Write (15,*)
      chtmp='Attractor     RES     Com_Ind     Open Space    No_Change'
      Write (15,*) chtmp
      Write (15,*)

      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      Subroutine    Integrtn(N_F, N_VAR)
      integer N_F, N_VAR, nn(10)
      character chtmp*55, SAS_NM(4)*12,crr*4,vcode*5
      real*8  x(4,N_F), U(4,N_VAR),a(10)
c
      data SAS_NM/'Est_RES.txt','Est_CI.txt','Est_OS.txt','Est_NCh.txt'/
c
      CALL OutFile()
c

      do i=1,4
         do j=1,N_F
            x(i,j)=0
        enddo
         do j=1,N_VAR
            U(i,j)=0
        enddo
      enddo
```

```
      do 235 i=1,4
         chtmp="Estimatn\"//SAS_NM(i)
         open(unit=i,file=chtmp, form='formatted')
         do LL=1,7
              Read (i,*)
         enddo

112           read(i,*, end=225) crr, (a(J),J=1,3)
         CALL Decode(crr,nn(1),nn(2))
         if (nn(2) .EQ. 1)    then
             x(i,nn(1))=a(2)
         else
             U(i,nn(1))=a(2)
         endif
         goto 112
225         close(i)

235      continue


        do LL=1,N_F
             CALL N_CHAR(LL, crr)
             vcode='X'//crr
             Write (15,350) vcode,(x(j,LL),j=1,4)
        enddo
        do LL=1,N_VAR
             CALL N_CHAR(LL, crr)
             vcode='U'//crr
             Write (15,350) vcode,(U(j,LL),j=1,4)
        enddo

350     format(2x,A5,3x,5(E12.6,2x))

      return
      end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      Subroutine Decode(ccc,n1,n2)
      character  ccc*4, ct(4)*1
      Integer n5, n1, n2, n3, nm(5)

        n5=len(ccc)


        do LL=1,n5
            ct(LL)=ccc(LL:LL)
            if (ct(LL) .NE. '') n3=LL
        enddo
c
        n1=0
```

```
      do LL=2,n3
           nm(5)=10**(n3-LL)
           nm(LL)=ichar(ct(LL))-48
           n1=n1+nm(LL)*nm(5)
      enddo

      n2=2
      if (ct(1) .EQ. 'x')  n2=1

     return
     end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     Subroutine N_CHAR(n5, ccc)
     character  ccc*4, ct(2)*1
     Integer n5, n1, n2
     n1=int(n5/10)
     n2=n5-10*n1
     ct(1)=CHAR(48+n2)
     ccc=ct(1)
     if (n1 .GT. 0) then
        ct(2)=CHAR(48+n1)
        ccc=ct(2)//ct(1)
     endif

     return
     end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     Subroutine DISCLM()

     print*,'*****************************************************'
     print*,'*                                         *'
     print*,'*              DISCLAIM                            *'
     print*,'*                                         *'
     print*,'*                                         *'
     print*,'*     The authors of this program are not responsible  *'
     print*,'*  to any computer hardware damage or information loss  *'
     print*,'*  due to improper use of this program.              *'
     print*,'*                                         *'
     print*,'*                                         *'
     print*,'*                                         *'
     print*,'*      ALL RIGHTS OF THIS PROGRAM ARE RESERVED        *'
     print*,'*                                         *'
     print*,'*                                         *'
     print*,'*****************************************************'
     print*,' '
     print*,' '
c
      pause
c
```

```
      do LK=1,10
        print*,' '
      enddo
c
c     print*,'*********************************************************'
c     print*,'                                          '
c     print*,'                                          '
c     print*,'            Integrating the estimation              '
c     print*,'                                          '
c     print*,'             of the weights of LEAM     '
c     print*,'                                          '
c     print*,'*********************************************************'
c     print*,' '
      do LK=1,5
        print*,' '
      enddo

      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*****************************************************************************
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 12-2005

2. REPORT TYPE Final

3. DATES COVERED (From - To)

4. TITLE AND SUBTITLE
Procedure for Estimating Weights of LEAM:
User's Guide

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Shoufan Fang, James Westervelt, George Z. Gertner, and Alan B. Anderson

5d. PROJECT NUMBER
622720A896

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Engineer Research and Development Center (ERDC)
Construction Engineering Research Laboratory (CERL)
PO Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
ERDC/CERL SR-05-59

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Strategic Environmental Research and
Development (SERDP) Office
Conservation Program Manager
901 N. Stuart Street, Suite 303
Arlington, VA 22203-1853

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

14. ABSTRACT

This document describes spatially explicit methodology to calibrate the Land use Evolution and impact Assessment Model (LEAM) using historic land use data. The LEAM model is used to predict urban growth and environmental impacts associated with urban growth. The model uses physical and social factors to simulate the dynamics of urban development. The Procedure for Estimating the Weights of LEAM (PEWL) uses logistic regression techniques and historic land use data to statistically calibrate the 13 subfactors used in LEAM. PEWL consists of computer programs to prepare samples, generate programs for regression analysis, and produce data files (estimated weights) for simulation using the Urban Sprawl Submodel of LEAM. The generated analysis programs from PEWL are executed using SAS, a common statistical package, to complete logistic estimation. PEWL was developed in FORTRAN language on Microsoft Developer Studio (Fortran PowerStation 4.0, 1993-1994) software. PEWL includes both the source and executable files of the FORTRAN programs. For demonstration purposes, the approach is demonstrated for a multi-county area around Fort Benning, GA.

15. SUBJECT TERMS

LEAM, urban growth	land use planning	simulation modeling
military training	environmental impact	encroachment

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER (include area code)
Unclassified	Unclassified	Unclassified	SAR	46	Alan B. Anderson / 217-373-7233